Security
Boundaries
IT SECURITY ACADEMY
www.SecAcademy.com

Security Boundaries

Introduction

Microsoft's Trustworthy Computing was launched in 2002 to ensure the security, confidentiality and reliability of data processing

The initiative defined the concept of a Windows security boundary.

Security Boundaries

Introduction

To qualify as a system security boundary, a system element has to meet the following requirements:

- The element must be security-critical: it has to be worth protecting
- The policy defining its operation must be specified clearly
- Even if the policy does not have to cover every possible data transfer method, every boundary violation possibility must be defined and evaluated

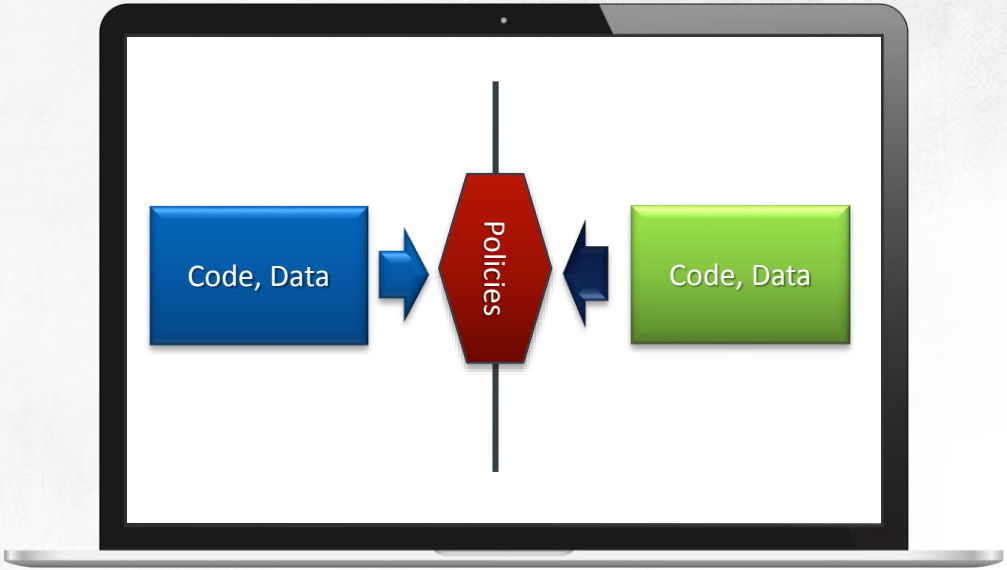
Security Boundaries

Introduction

Security boundaries are costly:

- They hinder a program's operation and may reduce the functionality of a system
- All boundary violations are considered equivalent to detecting a system vulnerability

There are fewer security boundaries in Windows than it would seem



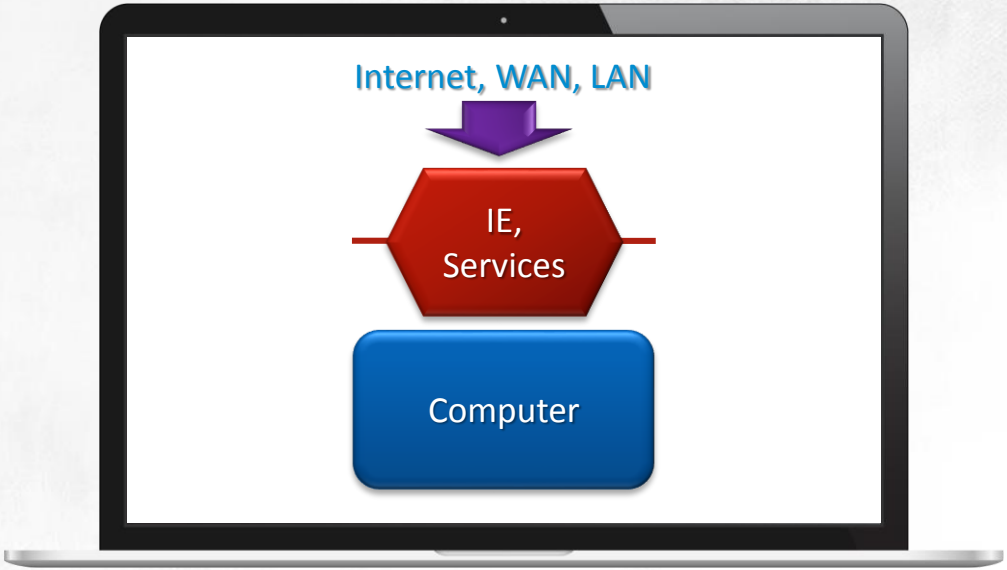
Three core and earliest security boundaries

Computer

The role of this security boundary, the physical computer, is to:

- Control the starting and running of programs whose source is not the computer itself (files from LAN, WAN and the Internet)
- Control all data sent and received by the programs

It does not offer protection from local attacks

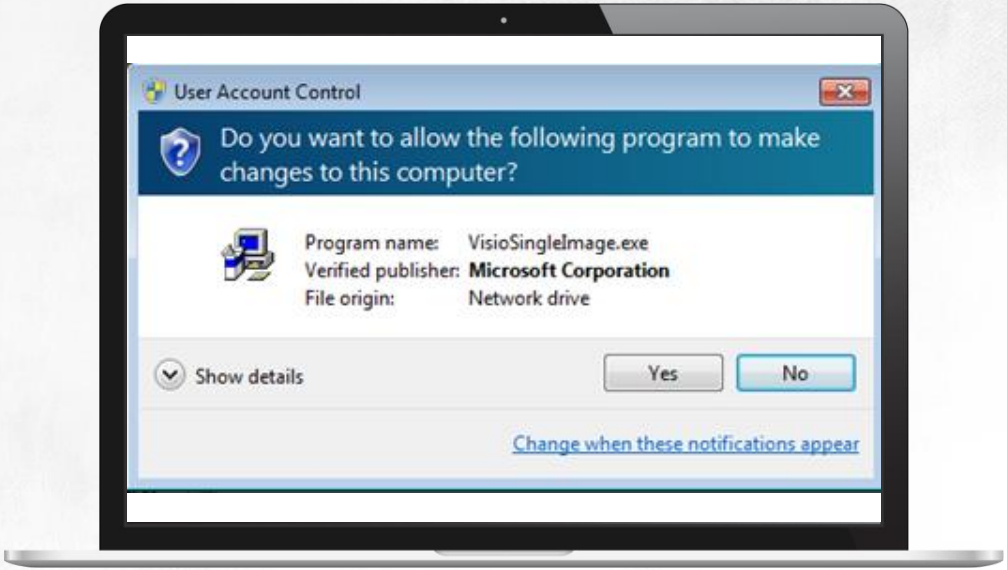


Three core and earliest security boundaries

Computer

Because processes that are running in a system can load data from the Internet, Windows systems as a rule do not trust programs and data that come from outside the local computer: until they can prove their authenticity, they are treated as malware

If this boundary is breached, a critical security update is released

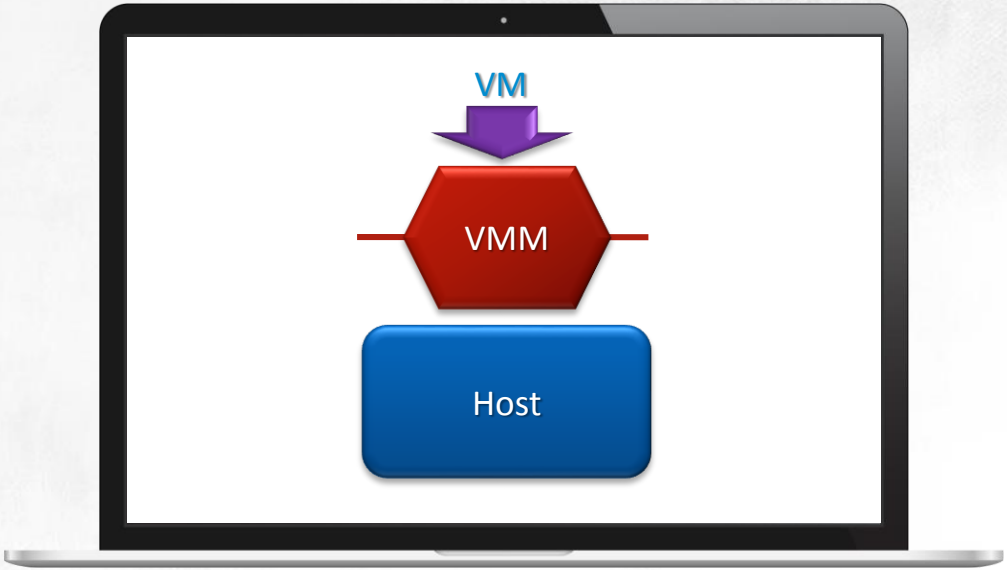

User Account Control
Do you want to allow the following program to make changes to this computer?
Program name: VisioSingleImage.exe
Verified publisher: Microsoft Corporation
File origin: Network drive
Show details
Yes
No
Change when these notifications appear

Three core and earliest security boundaries

Operating system

A process running in a virtual machine cannot uncontrollably read and modify data or run programs installed in the host's system and in other virtual machines

This task is handled by a VMM hypervisor



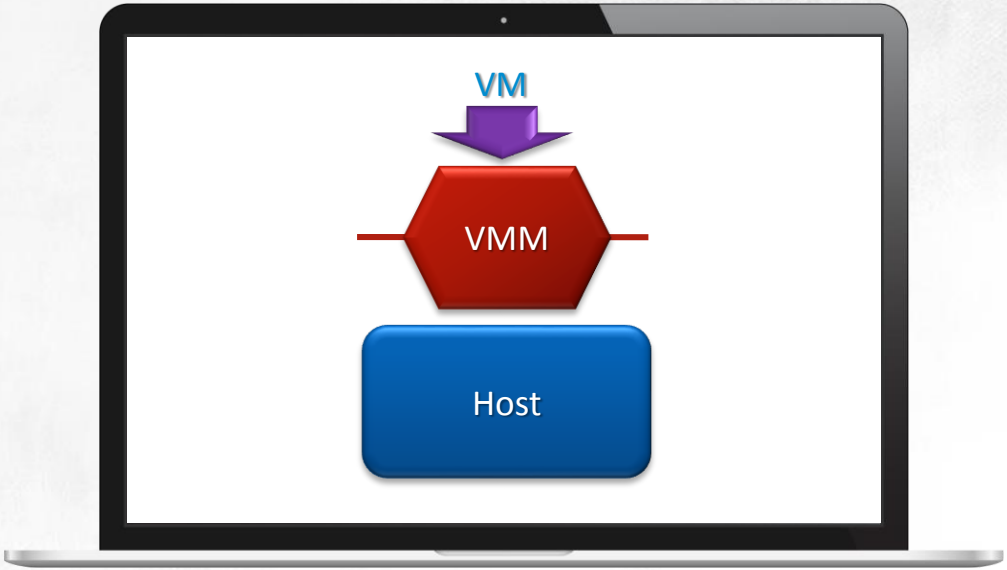
Three core and earliest security boundaries

Operating system

The first attacks that involved violating the virtual operating system boundary were noted in 1976 in IBM Systems Journal Paper (C.R. Attanasio's article)

Assuming that a host system, if protected, will ensure security for virtual systems running in it is a falsehood

Violating this boundary entails the release of an important security update

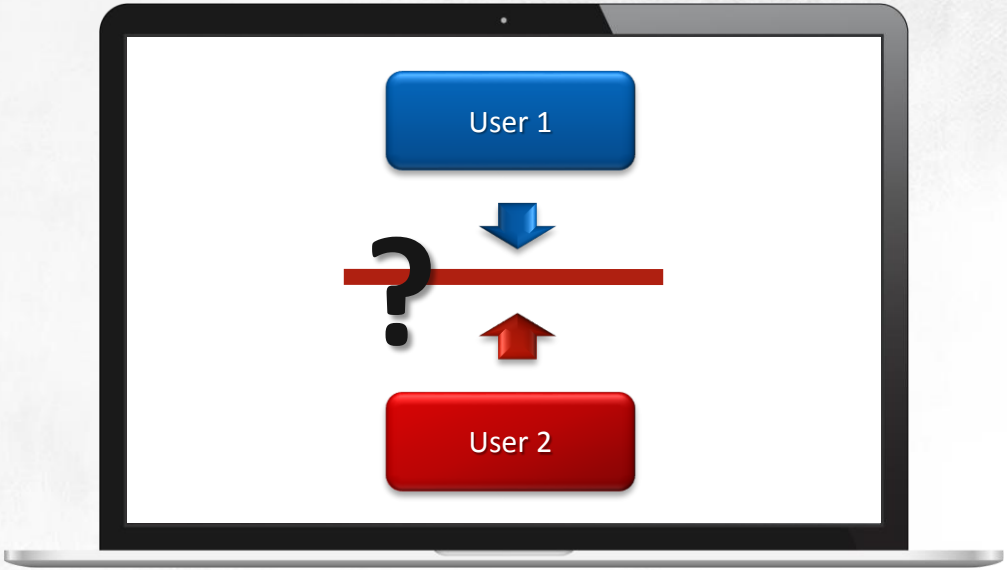


Three core and earliest security boundaries

User session

Windows separates user sessions and system sessions (session 0) as well isolates individual user sessions

- During system logon, each user is assigned the unique security identifier SID
- A SID is a piece of information included in every process a given user starts
- All objects created by these processes are placed in a private Based Name Object (BSO) namespace



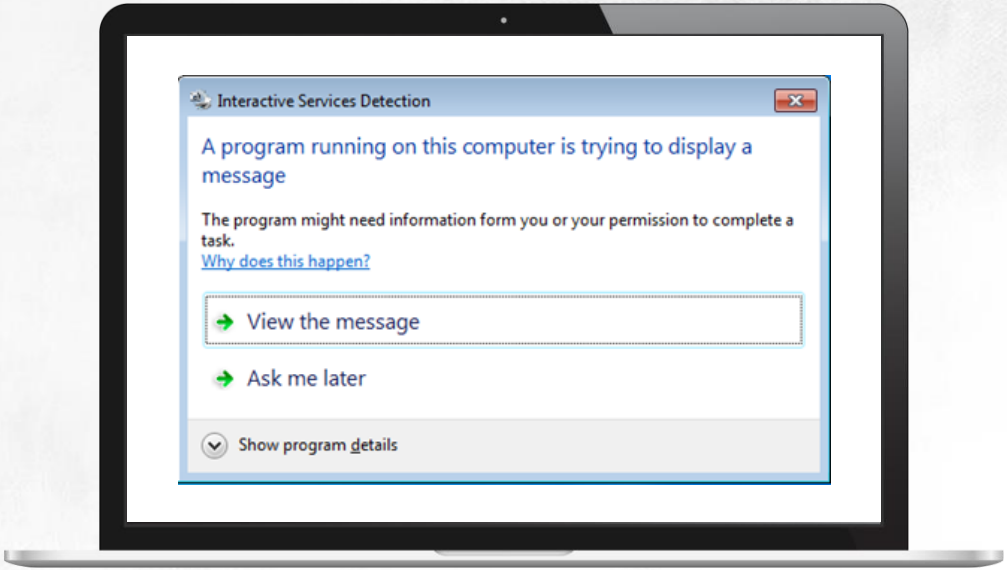
Three core and earliest security boundaries

User session

If not for the user session security boundary, Microsoft would not be able to secure terminal servers

On the other hand, if an administrator cannot manage user sessions, he can't control the computer

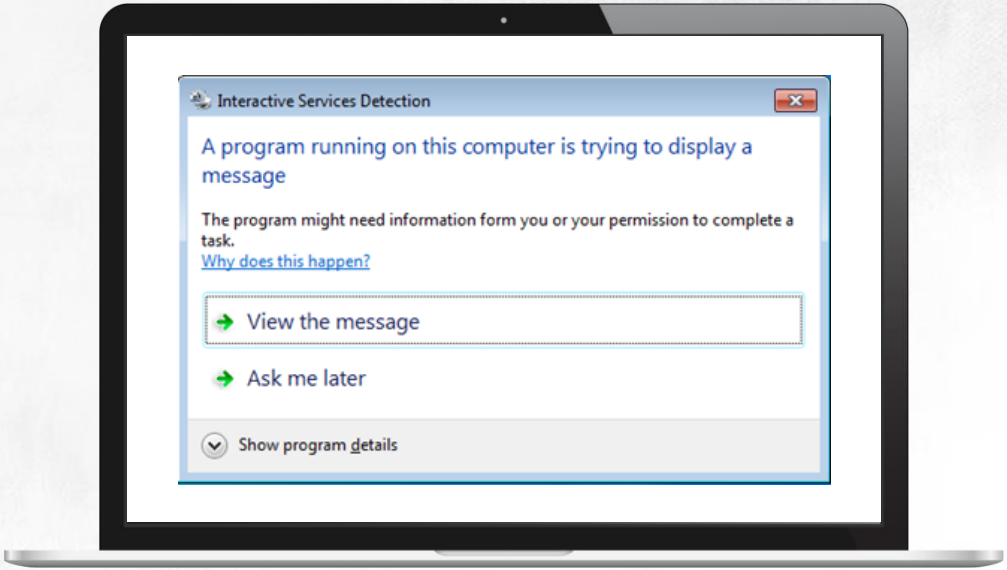
This is why the user session is only a security boundary for standard users in a system


Interactive Services Detection
A program running on this computer is trying to display a message
The program might need information form you or your permission to complete a task.
Why does this happen?
View the message
Ask me later
Show program details

Three core and earliest security boundaries

User session

Violating this security boundary entails the release of an important security update

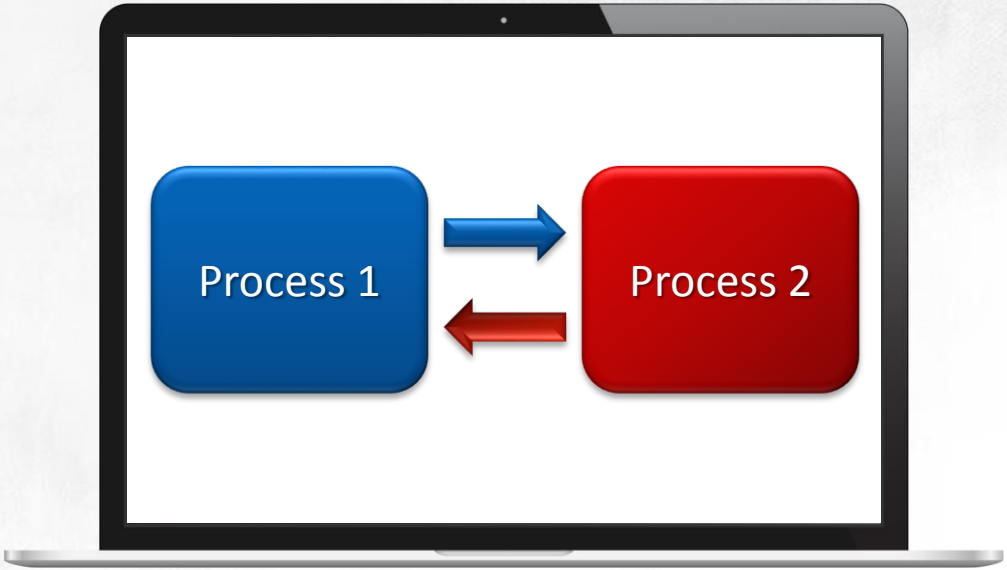

Interactive Services Detection
A program running on this computer is trying to display a message
The program might need information form you or your permission to complete a task.
Why does this happen?
View the message
Ask me later
Show program details

What a security boundary is not

Process isolation

A process is a container that includes:

- Binary code saved in an executable file
- Virtual memory assigned by the OS
- Handles, which make it possible to interact with it and control its operation
- Security context



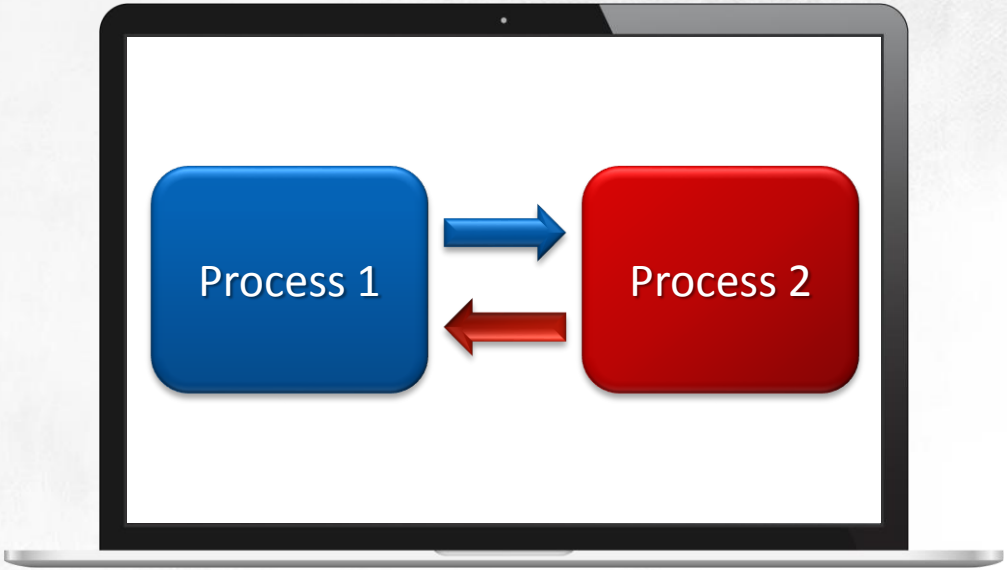
What a security boundary is not

Process isolation

A process defines a reliability boundary in Windows systems and is a unit for managing resources in an OS

However, a process can read and modify data of other processes launched by the same user and may change the way they operate

It is an application developer's decision to make it warn users against launching or connecting to an external process

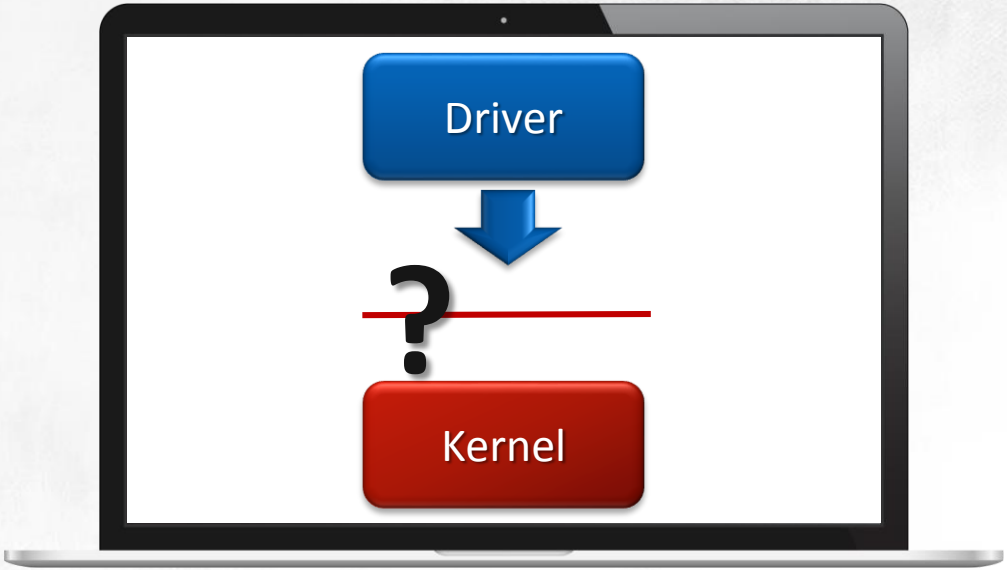


What a security boundary is not

PatchGuard

Installing and starting programs running in the kernel mode an administrator-exclusive operation. But if a process has been already started in this mode, the process will have uncontrolled access to every resource in a system

64-bit Windows systems protect the kernel from modification using the PatchGuard technology

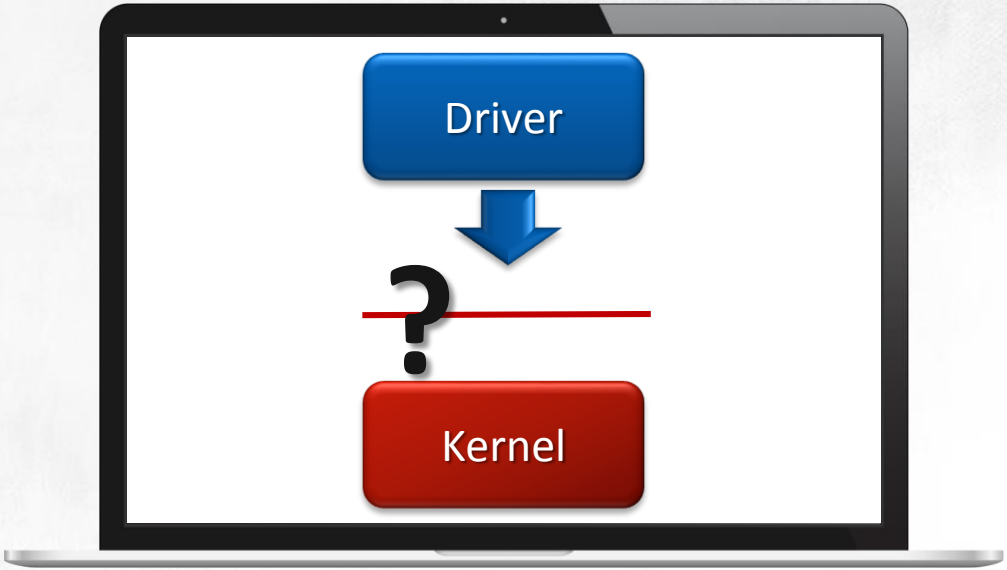


What a security boundary is not

PatchGuard

PatchGuard prevents patching the kernel (changing processes and structures of the kernel in a way not supported by Microsoft) by checking the signature of the most critical system processes like Ntoskrnl.exe, Hal.dll, IDT, SSDT and MSR

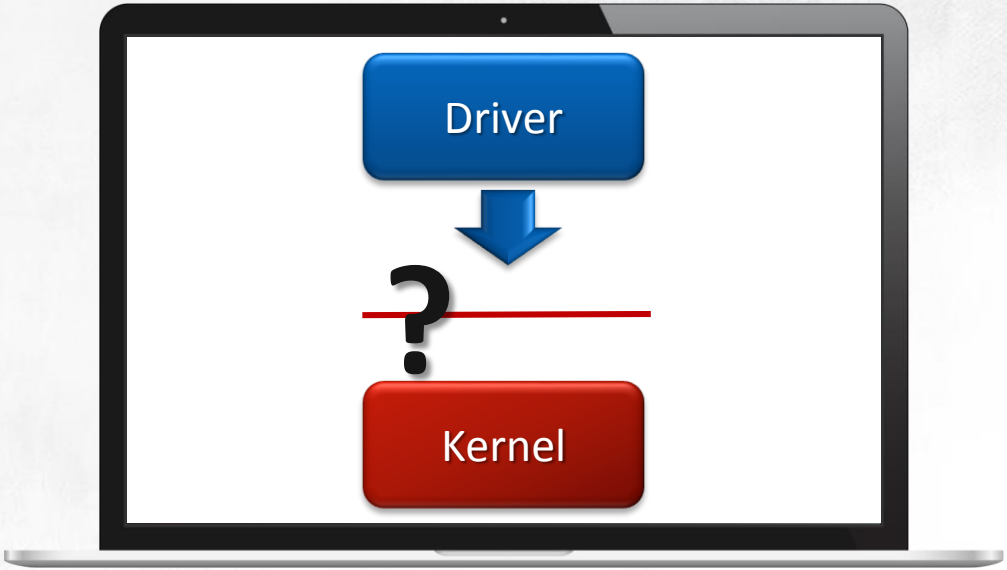
If PatchGuard detects they have been modified, it will stop the system



What a security boundary is not

PatchGuard

This mechanism for protecting the kernel is running with the same level of permissions as other kernel processes: PatchGuard has already been the target of several successful attacks (Bypassing PatchGuard on Windows x64)

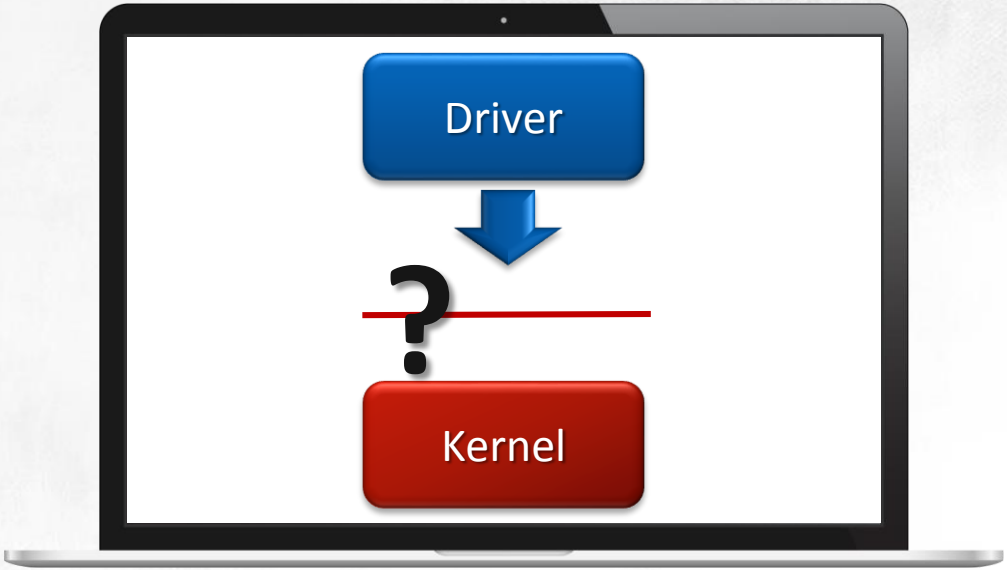


What a security boundary is not

PatchGuard

Not all processes and kernel structures are monitored: PatchGuard does not work against rootkits and viruses

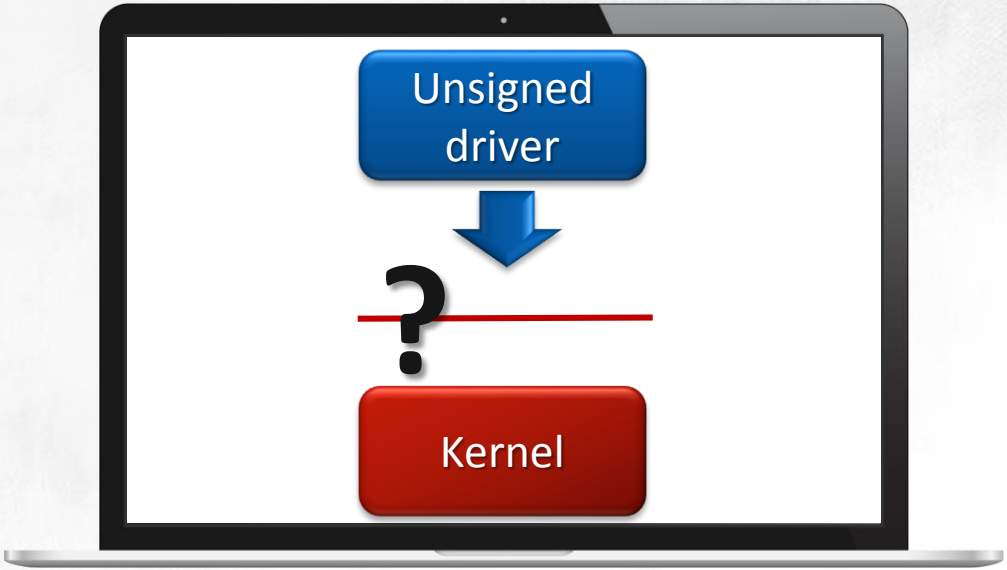
It also does not prevent attackers from modifying files stored on the system disk



What a security boundary is not

Kernel Mode Code Signing

Kernel Mode Code Signing is a technology available in 64-bit Windows systems. Its role is to make it harder for attackers to install faulty or malicious drivers and to facilitate the identification of driver manufacturers

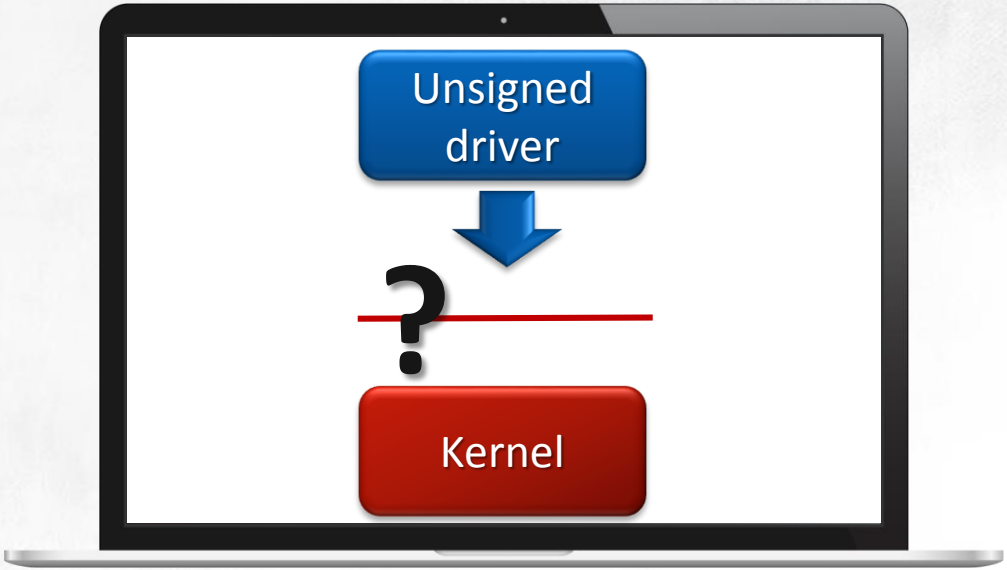


What a security boundary is not

Kernel Mode Code Signing

To be loaded and started, all drivers have to be digitally signed:

- At system start, a list of revoked and blocked drivers is loaded
- Before a driver is launched, its signature is verified. Additionally, the operating system checks if the certificate used for signing it was issued by a trusted authority and checks if it has been revoked



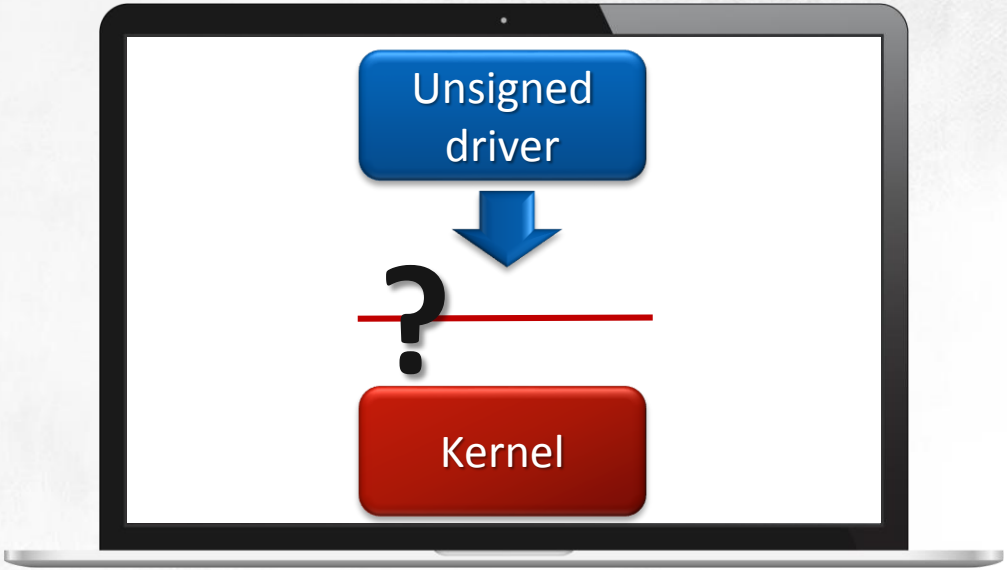
What a security boundary is not

Kernel Mode Code Signing

KMCS has also been attacked successfully several times (Black Hat 2007, Atsiv by Linchpin Labs)

Kernel Mode Code Signing cannot protect you from malicious drivers:

- Attackers only need to pay 300$ for a certificate
- It doesn't disable the modification of on-disk files

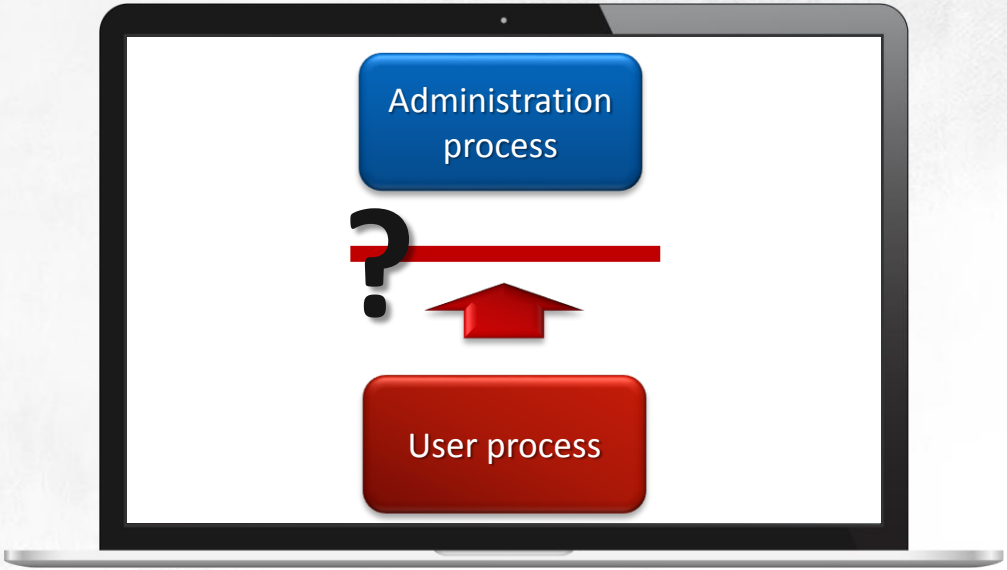


What a security boundary is not

User account control

User account control aims to ensure malware is not run with administrative permissions

However, it is still up to the user to make the decision whether to grant a program escalated permissions, and because most programs lack signatures, this decision is not fully informed and is made purely on the basis of the program's name

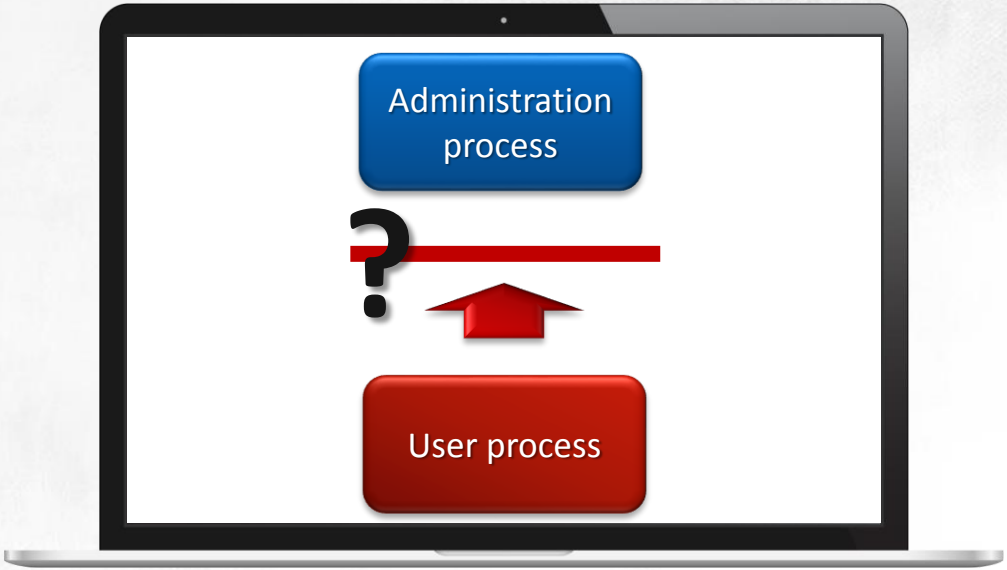


What a security boundary is not

User account control

Displaying the secure desktop, moreover, does not entail pressing the Ctrl+Alt+Del key sequence. You can read about lots of attacks online that switch users to a fake secure desktop

Finally, elevated processes are running in the same session and in the same namespace as processes having a lower level of permissions, meaning they are not isolated from each other

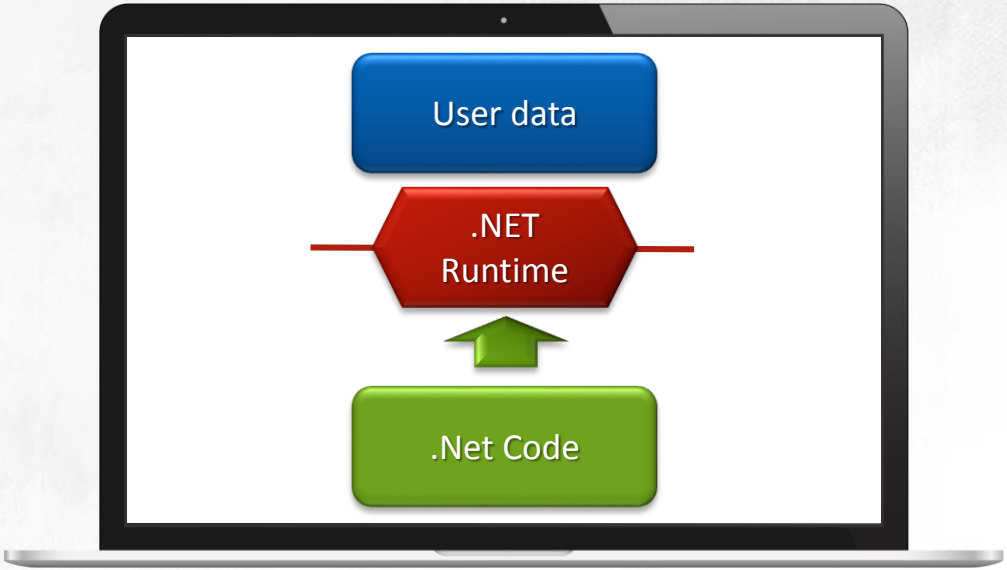


Java Virtual machine and code access security

The roles of both these mechanisms are to restrict the permissions programs have, and to isolate them in a way that's impossible to do at the process level

Code Access Security: a Java virtual machine and CAS in the .NET Runtime define and control program permissions independently from the privileges held by the users who run them

The resultant set of program policies depends on its trust level

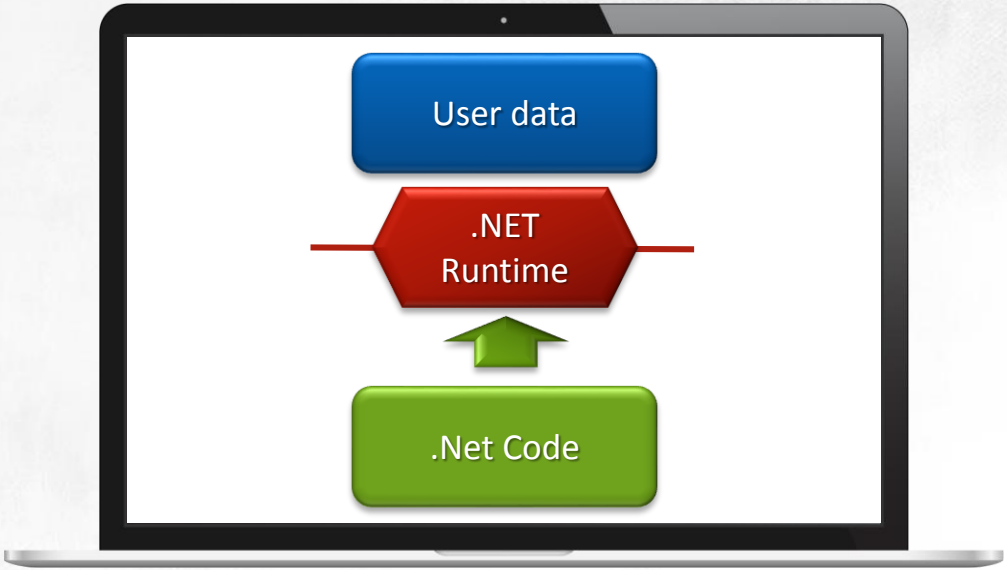


Java Virtual machine and code access security

When launching .NET, CLR checks its label and than compares the data the program requests with the permissions defined in the security policies set for this program

If the program will try to execute an operation without necessary permissions, an error will be reported

These mechanisms define a security boundary. Violating it entails the release of a critical security update

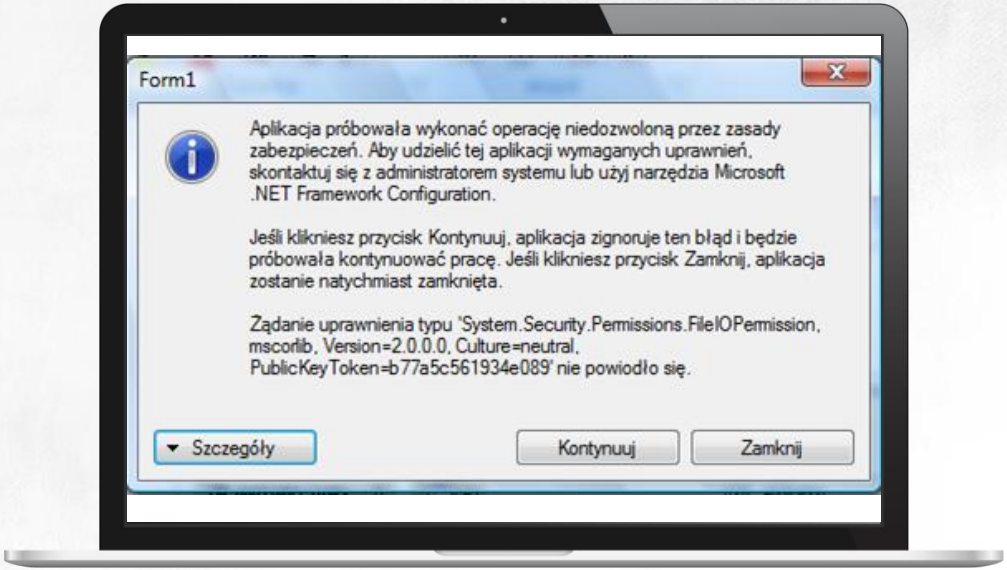


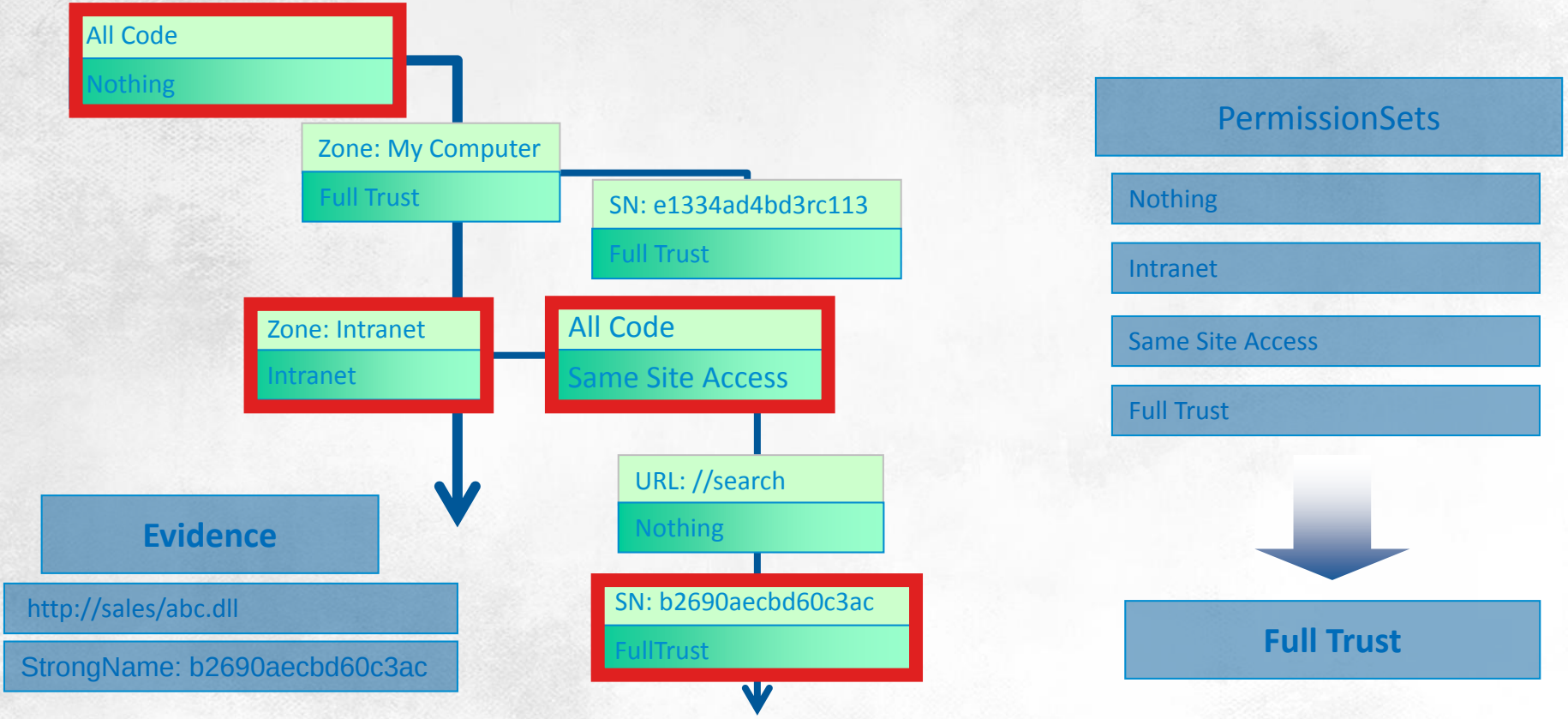
Java Virtual machine and code access security

When launching .NET, CLR checks its label and than compares the data the program requests with the permissions defined in the security policies set for this program

If the program will try to execute an operation without necessary permissions, an error will be reported

These mechanisms define a security boundary. Violating it entails the release of a critical security update


Form1
Aplikacja próbowała wykonać operację niedozwoloną przez zasady zabezpieczeń. Aby udzielić tej aplikacji wymaganych uprawnień, skontaktuj się z administratorem systemu lub użyj narzędzia Microsoft .NET Framework Configuration.
Jeśli klikniesz przycisk Kontynuuj, aplikacja zignoruje ten błąd i będzie próbowała kontynuować pracę. Jeśli klikniesz przycisk Zamknij, aplikacja zostanie natychmiast zamknięta.
Żądanie uprawnienia typu 'System.Security.Permissions.FileIOPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089' nie powiodło się.
Szczegóły
Kontynuuj
Zamknij


All Code
Nothing
Zone: My Computer
Full Trust
SN: e1334ad4bd3rc113
Full Trust
Zone: Intranet
Intranet
All Code
Same Site Access
URL: //search
Nothing
SN: b2690aecbd60c3ac
FullTrust
Evidence
http://sales/abc.dll
StrongName: b2690aecbd60c3ac
PermissionSets
Nothing
Intranet
Same Site Access
Full Trust
Full Trust

Code access security

Declarative security demand call

```
[FileIOPermission(SecurityAction.Demand,
   Read= "C:\\temp.txt")]
public string ReadTempFile()
{
// Code to read file goes here
}
```

Imperative security demand call

```
FileIOPermission filePerm = new FileIOPermission(
   FileIOPermissionAccess.Read, "C:\\temp.txt");
try
{
   filePerm.Demand();
   // Code to access file goes here
}
Catch (SecurityException e)
{
// if demand fails, this code is executed
}
```

THANKS